

计算机系统结构 学习指导与题解

尹朝庆 等 编著

- ◆ 知识要点阐释
- ◆ 题例分析解答
- ◆ 考研实战模拟

华中科技大学出版社



21 世纪计算机及相关专业课程学习辅导系列

计算机系统结构 学习指导与题解

尹朝庆 卢 苇
吴 钊 陈 波 编著

华中科技大学出版社

图书在版编目(CIP)数据

计算机系统结构学习指导与题解/尹朝庆 等编著
武汉:华中科技大学出版社, 2003年7月
ISBN 7-5609-2970-2

I. 计…

II. ①尹… ②卢… ③吴… ④陈…

III. 计算机体系结构-高等学校-教学参考资料

IV. TP303

计算机系统结构学习指导与题解

尹朝庆 等编著

策划编辑:沈旭日

封面设计:潘 群

责任编辑:叶见欣

责任校对:刘 飞

责任监印:熊庆玉

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87545012

录 排:华中科技大学惠友科技文印中心

印 刷:湖北省新华印务有限责任公司

开本:787×960 1/16

印张:16.75

字数:299 000

版次:2003年7月第1版

印次 2003年7月第1次印刷

定价:19.80元

ISBN 7-5609-2970-2/TP·507

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书旨在帮助读者加深对“计算机系统结构”课程知识要点的理解,掌握本课程的基本概念、基本理论和基本方法,提高分析问题和解决问题的能力。

本书归纳了“计算机系统结构”这门课程的主要内容和知识要点,较为深入地分析、研究了各章的重点与难点;通过对大量典型例题的深入剖析与详尽解答,帮助读者熟悉解题方法,掌握解题思路与技巧;提供了大量的练习题及其解答供读者作自我测评之用,最后一章给出了 3 套模拟试题和解答。附录汇集了全国部分重点大学的硕士研究生入学考试的试题。本书对讲授和学习“计算机系统结构”课程都会有很大的帮助,也可作为参加有关专业的研究生入学考试和其他考试的参考书。

本书可与华中科技大学出版社出版的“面向 21 世纪计算机专业本科系列教材”中的《计算机系统结构》配套使用。

前言

“计算机系统结构”是计算机科学与技术学科的一门重要的专业课程，主要研究计算机系统的顶层设计技术和高性能计算机的体系结构，涉及较多计算机学科的知识。迅速发展的高性能计算机是信息化的基础，计算机系统结构的发展是高性能计算机发展的主要推动力。计算机系统结构由低级向高级发展的过程，也就是计算机并行化不断发展的过程。“计算机系统结构”（或称为“计算机体系结构”）课程以计算机并行化的概念、方法与技术为主线，以性能评价方法为依托，讲授当前高性能计算机体系结构的主要技术和方法。学习本课程后，应能正确掌握计算机系统结构的基本概念，掌握计算机系统结构设计的基本思想和方法以及性能分析方法，了解目前比较成熟的高性能计算机的基本结构和使用的技术。

本课程的概念性和理论性较强，课程内容比较广泛，比较抽象，不易掌握。尤其在结构设计和性能的定量分析计算方面，需要灵活运用所学的基本理论和方法，才能正确地进行设计和分析计算。为了帮助学生更好地掌握计算机系统结构课程的内容，从传授知识和培养能力的目标出发，根据作者多年从事本课程教学的经验，阐述课程教学的特点、难点和知识要点，我们编写了本书。

本书以尹朝庆教授主编的华中科技大学出版社出版的“面向 21 世纪计算机专业本科系列教材”中的《计算机系统结构》为主要依据，参考了部分全国重点大学计算机科学与技术专业“计算机系统结构”课程的教学内容和国务院学位委员会组织编写的《同等学力人员申请硕士学位计算机科学与技术学科综合水平全国统一考试大纲与指南》的要求。全书分为 8 章。第 1 章是计算机系统结构导论，第 2 章是流水线技术与向量处理技术，第 3 章是存储系统，第 4 章是互连网络，第 5 章是单指令流多数据流计算机，第 6 章是多处理机系统，第 7 章是数据流计算机，第 8 章是模拟试卷与解答，部分模拟试题来源于重点大学硕士研究生入学考试的部分试卷。另外，附录汇集了部分全国重点大学硕士研究生入学考试的计算机系统结构试题。

为了便于学习，各章内容按基本要求与难点、主要内容与知识要点、典型例题分析与解答、测试题与解答四个部分进行组织，理清学习的知识要点、重点和



难点,帮助读者掌握各章的基本概念与基本理论、基本公式与基本方法。典型例题和测试题基本包括了尹朝庆主编的《计算机系统结构》的各章习题,并增加了部分其他题型的测试题。书中对例题、测试题和模拟考试题解答时重在解题方法和解题思路的阐述。读者可以根据这一部分内容对照检查自己的学习情况,测试自己的综合应用能力。

需要指出的是,计算机系统结构的基本概念和基本方法的应用是十分灵活的。因此,在“计算机系统结构”课程的学习过程中,读者不要急于看本书提供的解答,而应该首先领会教材内容和本书阐述的知识要点,对所学的基本原理和方法等理论性较强的内容进行归纳,掌握解题的思路和方法,独立完成习题,然后再与本书提供的解题范例进行比较。融会贯通后,才能加深对所学知识的理解和掌握,提高运用所学知识分析问题和解决问题的能力,提高解题的技巧。

参加本书编著的人员有尹朝庆、卢苇、吴钊和陈波,由尹朝庆主编和统稿。

本书可供计算机科学与技术专业的本科生、研究生以及从事计算机系统结构教学工作的教师参考,也可作为其他有关专业的教师和工程技术人员的参考书。

由于作者的水平和经验有限,难免有疏漏与不当之处,敬请读者批评指正。

作者

2002年10月

2002年10月

目 录

第 1 章 计算机系统结构导论	(1)
1.1 基本要求与难点	(1)
1. 本章基本要求	(1)
2. 本章的难点	(1)
1.2 主要内容与知识要点	(2)
1. 计算机系统的多级层次结构	(2)
2. 计算机系统结构、计算机组成与实现	(3)
3. 结构设计与软件可移植性	(3)
4. 计算机系统结构并行性的发展	(5)
5. 计算机系统的分类	(6)
6. 性能定量分析技术基础	(7)
7. 指令格式的优化设计	(9)
8. 按 CISC 方向发展与改进指令系统	(10)
9. 按 RISC 方向发展与改进指令系统	(12)
1.3 典型例题分析与解答	(12)
1.4 测试题	(19)
1.5 测试题解答	(25)
第 2 章 流水线技术与向量处理技术	(41)
2.1 基本要求与难点	(41)
1. 本章基本要求	(41)
2. 本章的难点	(42)
2.2 主要内容与知识要点	(42)
1. 指令重叠执行方式与先行控制技术	(42)
2. 流水执行方式及流水线分类	(43)
3. 流水线的性能分析与计算	(44)
4. 非线性流水线的调度	(45)
5. 流水线的相关及其处理方法	(47)



6. 多发射处理机及其性能分析	(49)
7. 向量处理方式与向量流水处理机的性能分析	(51)
2.3 典型例题分析与解答	(52)
2.4 测试题	(62)
2.5 测试题解答	(68)
第 3 章 存储系统	(81)
3.1 基本要求与难点	(81)
1. 本章基本要求	(81)
2. 本章的难点	(82)
3.2 主要内容与知识要点	(82)
1. 存储系统的基本概念	(82)
2. 并行存储器	(83)
3. 虚拟存储器的存储管理方式	(85)
4. 页面替换算法	(87)
5. Cache 存储器的地址映象与地址变换	(88)
6. Cache 存储器的替换算法及其实现	(90)
7. Cache 的一致性问题和解决方法	(91)
8. 三级存储系统的组织方式	(91)
3.3 典型例题分析与解答	(92)
3.4 测试题	(100)
3.5 测试题解答	(105)
第 4 章 互连网络	(118)
4.1 基本要求与难点	(118)
1. 本章基本要求	(118)
2. 本章的难点	(118)
4.2 主要内容与知识要点	(119)
1. 互连函数	(119)
2. 互连网络的结构参数和传输性能参数	(121)
3. 静态互连网络	(122)
4. 动态互连网络	(123)
5. 互连网络的消息传递机制	(127)
4.3 典型例题分析与解答	(130)
4.4 测试题	(138)

4.5 测试题解答	(144)
第 5 章 单指令流多数据流计算机	(159)
5.1 基本要求与难点	(159)
1. 本章基本要求	(159)
2. 本章的难点	(159)
5.2 主要内容与知识要点	(160)
1. SIMD 计算机的基本结构	(160)
2. SIMD 计算机的结构特点	(160)
3. ILLIAC IV 计算机系统	(161)
4. BSP 计算机	(161)
5. CM-2 计算机	(162)
6. MP-1 计算机系列	(162)
7. 阵列处理机上的差分计算	(163)
8. 阵列处理机的常用算法	(163)
5.3 典型例题分析与解答	(164)
5.4 测试题	(168)
5.5 测试题解答	(171)
第 6 章 多处理机系统	(179)
6.1 基本要求与难点	(179)
1. 本章基本要求	(179)
2. 本章的难点	(179)
6.2 主要内容与知识要点	(180)
1. 多处理机系统的基本结构	(180)
2. 多处理机的 Cache 一致性问题及产生原因	(181)
3. 监听协议	(181)
4. 基于目录的协议	(182)
5. CM-5 系统	(183)
6. SGI Origin 2000 系列服务器	(184)
7. 机群系统	(185)
8. 多机系统性能计算	(185)
6.3 典型例题分析与解答	(186)
6.4 测试题	(187)
6.5 测试题解答	(191)



第 7 章 数据流计算机	(199)
7.1 基本要求与难点	(199)
1. 本章基本要求	(199)
2. 本章的难点	(199)
7.2 主要内容与知识要点	(200)
1. 数据流计算机的两种驱动方式	(200)
2. 数据流计算机的指令结构及指令执行过程	(200)
3. 数据驱动方式的特点	(201)
4. 静态数据流计算机与动态数据流计算机	(201)
5. 数据流计算机的主要问题	(202)
6. 数据流程图	(203)
7.3 典型例题分析与解答	(205)
7.4 测试题	(206)
7.5 测试题解答	(210)
第 8 章 模拟试题与解答	(214)
模拟试卷(一)	(214)
模拟试卷(一)解答	(216)
模拟试卷(二)	(222)
模拟试卷(二)解答	(224)
模拟试卷(三)	(230)
模拟试卷(三)解答	(232)
附 录 全国重点大学硕士研究生入学考试试题汇集	(240)
清华大学硕士研究生入学考试试题 (一)	(240)
清华大学硕士研究生入学考试试题 (二)	(241)
清华大学硕士研究生入学考试试题 (三)	(242)
清华大学 1999 年硕士研究生入学考试试题	(243)
国防科技大学 1998 年硕士研究生入学考试试题	(246)
国防科技大学 2000 年硕士研究生入学考试试题	(250)
上海交通大学 1999 年硕士研究生入学考试试题	(253)
上海交通大学 2000 年硕士研究生入学考试试题	(255)
参考文献	(258)

第 1 章

计算机系统结构导论

1.1 基本要求与难点

本章主要介绍计算机系统结构的基本概念和计算机性能分析的基本知识,讨论机器指令字的优化设计方法,为深入学习后续各章内容打下基础。

1. 本章基本要求

- ① 领会通用计算机系统的多级层次结构及透明性概念。
- ② 掌握计算机系统结构、计算机组成和计算机实现的定义,以及三者之间的区别与联系。
- ③ 掌握实现软件移植的主要技术途径,领会软件兼容的定义以及系列机对软件兼容的基本要求;正确判断在系列机中发展新型号机器时,哪些改进是可取的。
- ④ 了解并行性的定义,领会并行性的二层含义和发展并行性的3种途径;了解系统耦合度的概念;了解Flynn分类法及其分类的典型机器的结构。
- ⑤ 领会指令平均周期数CPI和加速比S的概念掌握计算方法,掌握Amdahl定律及其用于性能改进的定量分析计算方法。
- ⑥ 熟练掌握定长码编码、哈夫曼编码和扩展编码的编码方法,掌握平均码长的计算方法,并能根据题目要求设计优化的指令格式。
- ⑦ 了解CISC存在的问题和RISC的优点,了解RISC机器设计的一般原则及基本技术,了解CISC与RISC相结合的发展趋势。

2. 本章的难点

- ① 应用Amdahl定律对性能改进情况进行定量分析计算。
- ② 操作码优化编码和指令字格式的优化设计。



1.2 主要内容与知识要点

1. 计算机系统的多级层次结构

(1) 层次结构的划分

计算机语言可以分成一系列的层次 (Level) 或级, 最低层语言的功能最简单, 最高层语言的功能最强, 更便于应用。从计算机语言的角度, 可将通用计算机系统在功能上划分成多级层次结构, 每一层以一种不同的语言为特征。按由低层到高层的顺序, 各层分别是: 微程序机器级、传统机器语言机器级、操作系统机器级、汇编语言机器级、高级语言机器级和应用语言机器级等。对每一层的使用者来说, 都可以把此机器级看成是一台独立的机器, 都可以应用相应的机器级语言。

微程序机器级和用组合逻辑控制的传统机器语言机器级都是用硬件实现的, 而采用微程序控制的传统机器语言机器级是用固件实现的。固件 (Firmware) 是一种具有软件功能的硬件, 例如, 把软件固化在只读存储器中就是一种固件。以硬件或固件为主实现的机器称为实际机器。

当前, 应用语言机器级、高级语言机器级、汇编语言机器级和操作系统机器级都是以软件为主实现的, 但也可根据用户对计算机性能/价格的不同要求增加硬件支持。以软件为主实现的机器称为虚拟机器 (Virtual Machine)。

(2) 透明性

从计算机系统的某一层的使用者角度看, 只需通过该层的语言就可以使用机器, 而不必关心其下层的机器级是如何工作和如何实现各自的功能, 这一性质称为透明性 (Transparency)。

计算机系统中的“透明”与通常意义下的“透明”在含义上正好相反。通常意义下的“透明”是指公开的, 什么都能看得到的, 例如, 玻璃是透明的, 那么, 可以看得玻璃瓶内的任何物体。计算机系统的“透明”是看不到的意思, 即对某一层的使用者来说, 他看不到该层以下各层的机器属性。

(3) 各机器级的实现技术

低层机器级为高层机器级的功能实现提供支持, 实现的技术是翻译和解释。翻译 (Translate) 是先用翻译程序 (Translator) 把高层机器级上的程序翻译成低层机器级上的等效程序, 然后在低层机器级上实现程序功能。解释 (Interpretation) 是用低层机器级的一些语句或指令来仿真高层机器级上的一条语句或指令的功能, 高层机器级程序中的每条语句或指令被逐条解释来实现程序功能。

采用翻译技术实现的典型例子有: 用翻译程序将应用语言机器级上的应用程序



包翻译成高级语言程序；用编译程序将高级语言源程序转换成机器语言目标程序；用汇编程序将汇编语言源程序转换成机器语言目标程序。采用解释技术实现的典型例子有：用解释方式在传统机器语言机器级上执行高级语言程序；用微指令程序解释实现机器指令；用微程序或机器指令程序解释实现操作系统的操作命令等。

2. 计算机系统结构、计算机组成与实现

(1) 计算机系统结构

计算机系统结构 (Computer Architecture) 也称为计算机体系结构，经典的计算机系统结构的定义是指计算机系统多级层次结构中传统机器语言机器级的结构，它是软件和硬件/固件的主要交界面，是由机器语言程序、汇编语言源程序和高级语言源程序翻译生成的机器语言目标程序能在机器上正确运行所应具有界面结构和功能。

目前，经常使用的是广义的计算机系统结构的概念，它除了包括经典的计算机系统结构的概念范畴外，还包括了计算机组成和实现技术。

(2) 计算机组成与实现

计算机组成是计算机系统结构的逻辑实现，它主要研究硬件系统在逻辑上是如何组织的，机器级内部数据流和控制流的组成与逻辑设计。计算机实现是指计算机组成的物理实现，主要着眼于器件技术和微组装技术。

(3) 结构、组成与实现之间的相互影响

相同结构的计算机可以因速度等性能要求不同而采用不同的组成，相同的组成可有多种不同的实现方法，这都取决于计算机系统的性能和价格要求以及器件技术的发展情况。

结构不同则可用的组成技术会有所不同，而不同的组成又会反过来影响系统结构的设计。因此，系统结构的设计不仅需要结合应用需求来考虑，要能为软件和算法的实现提供更多更好的硬件支持，同时要考虑可能采用和准备采用哪些组成技术，不能过多或不合理地限制各种可能的组成与实现技术的采用。

组成与实现可以折中权衡，主要取决于器件的来源和性能价格比。应该在可提供的器件及其实现技术的条件下，在使价格不增或少增的情况下尽可能提高系统的性能。

3. 结构设计与软件可移植性

软件可移植性 (Software Portability) 是指在一台机器上编制的软件不用修改或只经少量修改就可在另一台机器上运行，使得同一个软件可以应用于不同的硬件环境中。这样，在过去的计算机系统上使用的大量成熟可靠的软件，特别是应用软件，就可以在新机器上使用而不必重新编写。



在计算机系统结构的范畴内,实现软件可移植性的基本途径有:统一高级语言,采用系列机,以及模拟和仿真等。

(1) 统一高级语言

统一高级语言是指设计出一种对各种应用领域都比较高效的通用高级语言,在结构相同甚至完全不同的机器之间,通过配备不同的语言翻译程序就可以实现高级语言应用软件的移植。但是,不同的应用对高级语言的语法和语义结构的要求差别较大,使得难以实现一种统一通用的高级语言。从长远的目标来看,还是希望统一高级语言,但是,在目前相当一段时间内,只能是相对的统一。

(2) 系列机

系列机是指具有相同的系统结构,但具有不同组成和实现技术的一系列不同型号的机器。它们采用不同的组成和实现技术,从而具有不同的性能和价格,但从传统机器语言机器级的程序设计者角度来看,机器属性是相同的。

系列机技术既可以使同一系列的机器在汇编语言上实现统一,又能在同一系列内开发出性能价格比更高的新机器。因此,在结构相同或相近的机器之间能实现汇编语言应用软件和部分系统软件的移植。

软件兼容 (Software Compatibility) 是指同一个软件可以不加修改地运行于结构相同的各档机器上,而且运行结果一致。软件兼容有向上兼容、向下兼容、向前兼容、向后兼容之分。向上(下)兼容是指在某档机器上编制的程序可不加修改地运行于比它高(低)档的机器上;向前(后)兼容是指在某型号机器上编制的程序可不加修改地运行于在它之前(后)投入市场的机器上。对于系列机,必须保证做到向后兼容,力争做到向上兼容。

软件向后兼容是系列机的根本特征,也就是说,在先生产出的机器上编制的程序可以不加修改地在后生产的新机器上运行,而后生产的新机器增强了功能和提高了速度,从而提高了性能。因此,对于那些不属于系统结构设计,而属于计算机组成和实现的内容,不管是增加、删除,还是修改,都不会影响汇编语言程序和机器语言程序在系列机上的向后兼容。但是,对于属于系统结构设计范畴内的内容,为保证软件向后兼容,只能增加其新的功能或部件,而不能删除或更改已有的功能或部件。例如,可以把 CPU 与主存之间的数据通路的宽度由 16 位扩展到 32 位,以加速主机内部信息的传送;也可以把单总线改为双总线,以减少公共总线的使用冲突。这些内容都属于计算机组成的范畴,在逻辑设计时就要对它进行设计。例如,指令的功能设计与指令字格式设计和编码属于系统设计范畴,如果将原来全部采用定长操作码的指令改成操作码扩展编码,则虽然可以减小平均指令字长,但是,会直接导致以前编写的程序不能正确运行。

不同厂家生产的具有相同系统结构的计算机称为兼容机 (Compatible Machine)。



(3) 模拟与仿真

模拟与仿真能实现在结构不同的机器之间的机器语言程序的移植。模拟 (Simulation) 是用机器语言程序解释另一机器的机器指令来实现软件移植的。在两种机器的机器指令系统差异较大时, 使用模拟方法会使程序运行速度严重下降。仿真 (Emulation) 是用微程序直接解释另一机器的机器指令来实现软件移植的。采用仿真方法可以提高被移植软件的运行速度, 但在机器结构差异较大情况下, 难以实现仿真。因此, 在不同系列机器之间实现软件移植时, 可将模拟和仿真两种技术结合起来使用, 对于频繁使用且容易仿真的这部分机器指令采用仿真的方法, 以提高被移植软件的运行速度; 对于较少使用、对速度要求不高且难以仿真的这部分指令及 I/O 操作采用模拟的方法。

模拟与仿真的区别是: 模拟是用机器语言程序解释指令, 其解释程序存储在主存中; 仿真用微程序解释指令, 其解释程序存储在控制存储器中。

4. 计算机系统结构并行性的发展

(1) 并行性

并行性 (Parallelism) 是指同一时刻或同一时间间隔内发生两种或两种以上性质相同或不不同的事件。只要时间上相互重叠, 就存在并行性。并行性实际上包含同时性与并发性两层含义。同时性 (Simultaneity) 是指两个或多个事件在同一时刻发生。并发性 (Concurrency) 是指两个或多个事件在同一时间间隔内发生。

(2) 提高并行性的技术途径

在计算机系统中, 提高并行性的技术途径可以归纳为如下 3 种。

① 时间重叠 (Time Inteleaving): 在并行性概念中引入时间因素, 让多个处理过程在时间上错开, 轮流重叠地使用同一套硬件设备的各个部分, 加快硬件使用的周转来提高处理速度。典型的例子是流水技术。

② 资源重复 (Resource Replication): 在并行性概念中引入空间因素, 通过重复设置资源, 尤其是硬件资源, 来提高系统的性能。典型的例子有双工系统、相联处理机、阵列处理机等。

③ 资源共享 (Resource Sharing): 利用软件方法让多个任务按一定顺序轮流使用一套资源, 通过提高系统资源利用率来提高系统的性能和效率。典型的例子有多道程序分时系统、计算机网络和分布处理系统等。

从单处理机向现代高性能的多处理机系统发展的过程来看, 按时间重叠途径, 将多个处理机按宏流水技术构成的多处理机系统, 一般都是非对称型 (Asymmetrical) 或异构型 (Heterogeneous) 多处理机系统 (Multiprocessor System); 按资源重复途径构成的相联处理机系统和阵列处理机系统都是对称型 (Symmetrical) 或同构型 (Homogeneous) 多处理机系统; 按资源共享途径发展



的多处理系统可以是同构型的,也可以是异构型的。

20 世纪 90 年代以来,计算机系统最主要的发展是开发大规模并行处理(MPP)系统,其中,多处理机系统和多计算机系统是研究和开发的热点,它们可以由数十个乃至数千个微处理机构成的 MPP 系统,也可以是由多个高性能工作站或高档微机使用高速通信网络互连来构成高效并行处理的机群(Cluster)系统。这是因为当前重大科学计算课题要求计算机系统能达到 3T 性能,即 1 TFLOPS 的计算能力,1 TB 的主存容量和 1 TB/s 的 I/O 带宽。

(3) 多机系统的耦合度

耦合度用来反映多机系统中各处理机(计算机)之间物理连接的紧密程度和交互作用能力的强弱程度。多机系统的耦合度可分为最低耦合、松散耦合和紧密耦合等 3 种。

最低耦合系统(Least Coupled System)是指计算机之间没有物理的连接,最多只是通过盘、带等存储介质对机器之间的交互作用提供支持的系统。松散耦合系统(Loosely Coupled System)是指计算机或处理机之间通过低速通信线路互连或通过通道互连,共享一部分外设的系统。紧密耦合系统(Tightly Coupled System)是指处理机之间通过高速总线或开关网络互连,共享主存的系统。

5. 计算机系统的分类

(1) 指令流与数据流

指令流(Instruction Stream)是指机器执行的指令序列。数据流(Data Stream)是指由指令流调用的数据序列,包括输入数据和中间结果数据。

(2) 多倍性

多倍性(Multiplicity)是指在系统最受限制的部件上,同时处于同一执行阶段的指令或数据的最大可能个数。

(3) 弗林分类法

弗林(Flynn)分类法按指令流和数据流的多倍性把计算机系统分为单指令流单数据流(SISD)系统、单指令流多数据流(SIMD)系统、多指令流单数据流(MISD)系统和多指令流多数据流(MIMD)系统等 4 类。

传统单处理机属于 SISD 系统。阵列处理机和流水线单处理机属于 SIMD 系统。宏流水和脉动阵列机属于 MISD 系统。能全面实现作业、任务、指令、数组等各级并行的多处理机系统和多计算机系统都属于 MIMD 系统。

Flynn 分类法不适用于现代非控制流计算机系统的分类。可以根据执行程序或指令的控制方式对计算机系统分类,从而将其分为:由控制驱动的控制流方式,由数据驱动的数据流方式,由需求驱动的归约方式以及由模式驱动的匹配方式等计算机系统。



6. 性能定量分析技术基础

(1) CPU 性能

程序在机器上运行所需的时间:

$$\begin{aligned}\text{CPU 时间} &= \text{CPU 运行时钟周期数} \times \text{时钟周期} \\ &= \text{CPU 运行时钟周期数} / \text{时钟频率}\end{aligned}$$

若程序运行时执行的指令条数为 IC , 则每条指令的平均时钟周期数

$$CPI = \text{CPU 运行时钟周期数} / IC$$

由此可得出CPU性能公式:

$$\text{CPU 时间} = (IC \times CPI) / \text{时钟频率}$$

CPU性能公式的3个参数反映了与系统结构相关的3个因素: 时钟频率取决于计算机实现技术和计算机组织, 机器指令的平均时钟周期数 CPI 取决于计算机指令集的结构和指令集的设计与实现技术, 程序使用的指令条数 IC 取决于计算机指令集的结构和编译技术。通过改进计算机系统设计, 可以相应提高这3个参数的指标, 从而提高CPU性能。

假设计算机指令集有 n 种指令, 其中, 指令 i 的执行时间为 CPI_i , 若程序中指令 i 出现的次数为 IC_i , 则程序执行所需的时间:

$$\text{CPU 时间} = \sum_{i=1}^n (CPI_i \times IC_i) / \text{时钟频率}$$

指令的平均时钟周期数

$$\begin{aligned}CPI &= \sum_{i=1}^n (CPI_i \times IC_i) / IC \\ &= \sum_{i=1}^n (CPI_i \times IC_i / IC)\end{aligned}$$

其中, IC_i / IC 为指令 i 在程序中出现的概率, 或称为使用频度。

(2) Amdahl 定律

Amdahl定律用加速比 (Speedup Ratio) 来衡量系统采用的改进措施对系统性能提高的程度, 加速比

$$S_n = T_o / T_n$$

其中, T_o 为改进前所需的执行时间; T_n 为改进后所需的执行时间。

若 F_e 表示可被改进部分改进前的执行时间相对 T_o 的百分比, S_e 表示可被改进部分改进前的执行时间与改进后的执行时间的比值, 即被改进部分经改进后速度提高的倍数, 那么, 改进后整个系统的加速比

$$S_n = T_o / T_n = \frac{1}{(1 - F_e) + F_e / S_e}$$